**GOLD API DEVELOPMENT: Python (Django + REST + Pycharm)**

Prerequisites:

1. Set up a new Python project, together with its virtual environment (the usual practice for any Python code)
2. Pip install both Django and Djangorestframework
3. Inside this Python project, create a new Django project by typing ”*django-admin*

*startproject* + project name” in the terminal (which should be already positioned in the project folder)

1. Inside this Django project (access it with the terminal with *cd*), create an app with *“django-admin startapp* + name”
2. Sync your database for the first time by running *python manage.py migrate*. Django uses databases as its backbone.
3. Create your own superuser with *python manage.py createsuperuser*
4. Inside the project settings file, add 'rest\_framework' to the list of installed apps
5. *GET CODING!*

In this same folder you can find a basic API application, created following the tutorial at <https://www.django-rest-framework.org/tutorial/1-serialization/>. It presents the usual (and recommended) architecture for these projects, together with comments explaining its particular aspects. As a quick overview, let’s review the principal files:

* **Models**: The backbone of the API, the shape of the data to be used through the API is defined through python classes.
* **Serializers**: Establishes the link between the models and the API, producing accessible tables and lists.
* **Views**: Here the auxiliary classes to be used are defined. It is strongly advised to also implement any functions as classes, to both make callings more accessible and to make better use of the Django REST syntax.
* **Urls**: Links each of the views to a URL.
* **Permissions**: Optional file where custom permissions are established.

It must be taken into account that the presented code is very Django REST heavy, in that it uses a lot of its custom syntax as to code as efficiently as possible. All the documentation regarding it can be found in [*https://www.django-rest-framework.org/*](https://www.django-rest-framework.org/api-guide/) . Most files also provide “extended” versions of the code used, where more general syntax is used.